CS 475/575

Chih Hsuan Huang

ID: 934554197

huanchih@oregonstate.edu

Project #5

CUDA: Monte Carlo Simulation

1. Tell what machine you ran this on

   rabbit serve

2. What do you think this new probability is?

   The actual probability is 83.77%. This is calculated by choosing a maximum number of 256 threads of 2097152.

3. Show the rectangular table and the two graphs

Number of Trials	BlockSize	MegaTrials/Second	Probability
1024	8	12.36	83.5
1024	32	12.7847	85.25
1024	64	12.8308	84.18
1024	128	7.5206	82.42
1024	256	12.7745	85.16
4096	8	44.8179	83.69
4096	32	41.1443	83.76
4096	64	51.7799	84.47
4096	128	50.2947	83.47
4096	256	43.493	84.23
16384	8	147.1264	83.14
16384	32	190.9023	83.79
16384	64	168.9769	83.78
16384	128	198.6806	84.11
16384	256	129.6531	83.61
65536	8	412.7368	83.89
65536	32	448.5326	83.91
65536	64	658.3092	84.12
65536	128	675.9076	84.07
65536	256	553.6632	83.86
262144	8	693.1799	83.86
262144	32	1621.2151	83.78
262144	64	1568.7476	83.79
262144	128	1663.0127	83.96
262144	256	1359.4425	83.88
1048576	8	808.7669	83.81
1048576	32	2362.1683	83.8
1048576	64	2404.8143	83.83
1048576	128	2714.6052	83.85
1048576	256	2718.2081	83.8
2097152	8	830.0109	83.82
2097152	32	2785.9208	83.86
2097152	64	2931.4726	83.82
2097152	128	2986.7832	83.82
2097152	256	3068.5958	83.77

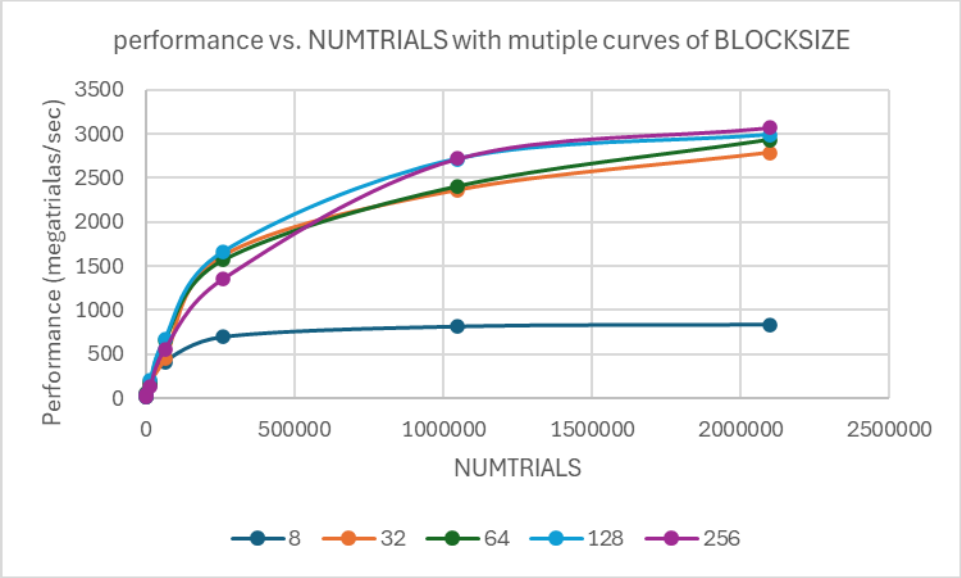

Figure 1:

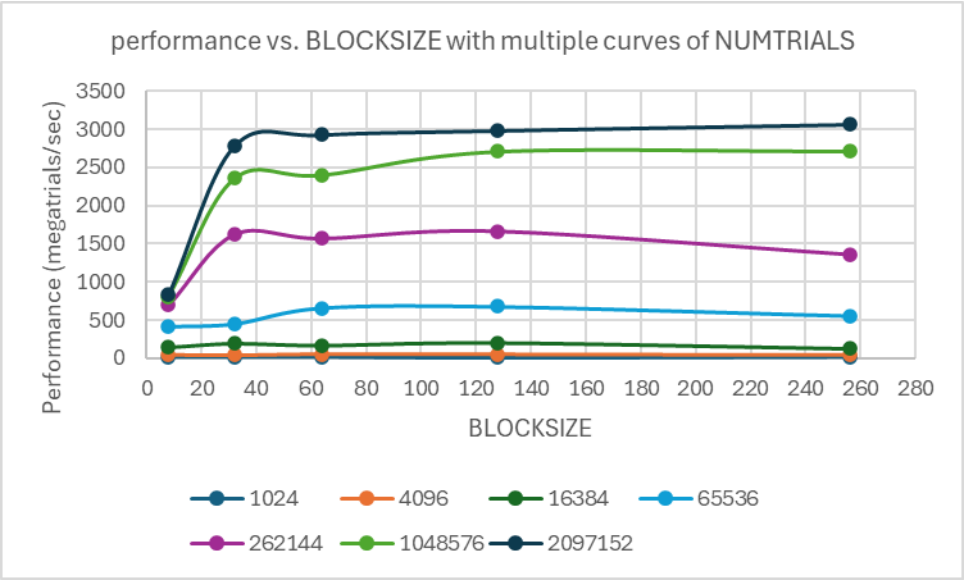

Figure 2:

4. What patterns are you seeing in the performance curves?

Figure 1:

As NUMTRIALS increases, performance improves significantly. The performance gains level off when NUMTRIALS reaches about 1048576. Larger BLOCKSIZE (above 32) performs better than smaller BLOCKSIZE (such as 8). A larger BLOCKSIZE has more outstanding performance when the number of tests is larger.

Figure 2:

As BLOCKSIZE increases, performance improves rapidly. Performance peaks when BLOCKSIZE reaches about 64. After reaching the peak, performance levels off or drops slightly. Larger NUMTRIALS perform better than smaller NUMTRIALS at all BLOCKSIZEs.

5. Why do you think the patterns look this way?

When NUMTRIALS increases, the GPU can better utilize parallel computing capabilities. More trials result in higher utilization of GPU resources, and each processor should be allocated more blocks. However, as the amount of data grows, the resources are fully utilized and the performance improvement becomes less obvious. As BLOCKSIZE increases, more computing units are utilized and performance improves. However, an excessively large BLOCKSIZE will cause increased thread management and synchronization overhead, thus affecting performance. And a smaller BLOCKSIZE (such as 8) causes discontinuous memory access, reduces memory access efficiency, and affects overall performance.

6. Why is a BLOCKSIZE of 8 so much worse than the others?

The computing units in the GPU are scheduled in "warp" units, and each warp contains 32 threads. When BLOCKSIZE is 8, a warp cannot be fully filled, causing computing resources to be underutilized, thus affecting performance. The other three block sizes, 32, 64 and 128, are all multiples of 32. This allows these larger block sizes to be fully utilized for maximum performance.

7. How do these performance results compare with what you got in Project #1? Why?

In Project 1, we performed a Monte Carlo simulation using OpenMP, and the performance using CUDA in Project 5 was

significantly higher than that using OpenMP in Project 1. CUDA is a parallel computing platform and API model created by NVIDIA. Because the parallel computing capability of GPU far exceeds the multi-thread processing capability of CPU, especially when processing a large number of computing tasks

8. What does this mean for what you can do with GPU parallel computing?

When we pursue peak performance, we can take full advantage of the parallel computing power of the GPU. Increasing the amount of data improves performance because it keeps all threads working and preventing them from being idle. We usually choose a multiple of 32 for the data size, which ensures a full thread queue for each calculation block. However, the block size can only be increased up to a certain point. Beyond this limit, further increasing the block size will not only fail to improve performance, but may actually decrease it. This is because overly large blocks increase thread management and synchronization overhead, affecting overall performance. Therefore, rational selection of block size and data volume, and adjustment and optimization for specific GPU architectures are key to fully utilizing the parallel computing capabilities of GPUs.